

```
In [1]: from sklearn.neighbors import NearestNeighbors, KNeighborsClassifier, KNeighbors
```

```
In [2]: from sklearn.datasets import load_iris
```

```
In [3]: iris_data=load_iris()
```

```
In [4]: iris_data.data
```

```
[6.6, 2.9, 4.6, 1.3],
[5.2, 2.7, 3.9, 1.4],
[5. , 2. , 3.5, 1. ],
[5.9, 3. , 4.2, 1.5],
[6. , 2.2, 4. , 1. ],
[6.1, 2.9, 4.7, 1.4],
[5.6, 2.9, 3.6, 1.3],
[6.7, 3.1, 4.4, 1.4],
[5.6, 3. , 4.5, 1.5],
[5.8, 2.7, 4.1, 1. ],
[6.2, 2.2, 4.5, 1.5],
[5.6, 2.5, 3.9, 1.1],
[5.9, 3.2, 4.8, 1.8],
[6.1, 2.8, 4. , 1.3],
[6.3, 2.5, 4.9, 1.5],
[6.1, 2.8, 4.7, 1.2],
[6.4, 2.9, 4.3, 1.3],
[6.6, 3. , 4.4, 1.4],
[6.8, 2.8, 4.8, 1.4],
[6.7, 3. , 5. , 1.7],
```

```
In [5]: iris_data.target
```

```
Out[5]: array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
               0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
               0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
               1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
               1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2,
               2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
               2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

```
In [6]: from sklearn.model_selection import cross_val_predict
```

```
In [7]: x=iris_data.data
        y=iris_data.target
        knn = KNeighborsClassifier(n_neighbors=10)
        y_pred= cross_val_predict(knn, x, y, cv=5)
```



```
In [16]: boston_data.data
```

```
Out[16]: array([[6.3200e-03, 1.8000e+01, 2.3100e+00, ..., 1.5300e+01, 3.9690e+02,  
                4.9800e+00],  
               [2.7310e-02, 0.0000e+00, 7.0700e+00, ..., 1.7800e+01, 3.9690e+02,  
                9.1400e+00],  
               [2.7290e-02, 0.0000e+00, 7.0700e+00, ..., 1.7800e+01, 3.9283e+02,  
                4.0300e+00],  
               ...,  
               [6.0760e-02, 0.0000e+00, 1.1930e+01, ..., 2.1000e+01, 3.9690e+02,  
                5.6400e+00],  
               [1.0959e-01, 0.0000e+00, 1.1930e+01, ..., 2.1000e+01, 3.9345e+02,  
                6.4800e+00],  
               [4.7410e-02, 0.0000e+00, 1.1930e+01, ..., 2.1000e+01, 3.9690e+02,  
                7.8800e+00]])
```

```
In [17]: boston_data.target
```

```
Out[17]: array([24. , 21.6, 34.7, 33.4, 36.2, 28.7, 22.9, 27.1, 16.5, 18.9, 15. ,
 18.9, 21.7, 20.4, 18.2, 19.9, 23.1, 17.5, 20.2, 18.2, 13.6, 19.6,
 15.2, 14.5, 15.6, 13.9, 16.6, 14.8, 18.4, 21. , 12.7, 14.5, 13.2,
 13.1, 13.5, 18.9, 20. , 21. , 24.7, 30.8, 34.9, 26.6, 25.3, 24.7,
 21.2, 19.3, 20. , 16.6, 14.4, 19.4, 19.7, 20.5, 25. , 23.4, 18.9,
 35.4, 24.7, 31.6, 23.3, 19.6, 18.7, 16. , 22.2, 25. , 33. , 23.5,
 19.4, 22. , 17.4, 20.9, 24.2, 21.7, 22.8, 23.4, 24.1, 21.4, 20. ,
 20.8, 21.2, 20.3, 28. , 23.9, 24.8, 22.9, 23.9, 26.6, 22.5, 22.2,
 23.6, 28.7, 22.6, 22. , 22.9, 25. , 20.6, 28.4, 21.4, 38.7, 43.8,
 33.2, 27.5, 26.5, 18.6, 19.3, 20.1, 19.5, 19.5, 20.4, 19.8, 19.4,
 21.7, 22.8, 18.8, 18.7, 18.5, 18.3, 21.2, 19.2, 20.4, 19.3, 22. ,
 20.3, 20.5, 17.3, 18.8, 21.4, 15.7, 16.2, 18. , 14.3, 19.2, 19.6,
 23. , 18.4, 15.6, 18.1, 17.4, 17.1, 13.3, 17.8, 14. , 14.4, 13.4,
 15.6, 11.8, 13.8, 15.6, 14.6, 17.8, 15.4, 21.5, 19.6, 15.3, 19.4,
 17. , 15.6, 13.1, 41.3, 24.3, 23.3, 27. , 50. , 50. , 50. , 22.7,
 25. , 50. , 23.8, 23.8, 22.3, 17.4, 19.1, 23.1, 23.6, 22.6, 29.4,
 23.2, 24.6, 29.9, 37.2, 39.8, 36.2, 37.9, 32.5, 26.4, 29.6, 50. ,
 32. , 29.8, 34.9, 37. , 30.5, 36.4, 31.1, 29.1, 50. , 33.3, 30.3,
 34.6, 34.9, 32.9, 24.1, 42.3, 48.5, 50. , 22.6, 24.4, 22.5, 24.4,
 20. , 21.7, 19.3, 22.4, 28.1, 23.7, 25. , 23.3, 28.7, 21.5, 23. ,
 26.7, 21.7, 27.5, 30.1, 44.8, 50. , 37.6, 31.6, 46.7, 31.5, 24.3,
 31.7, 41.7, 48.3, 29. , 24. , 25.1, 31.5, 23.7, 23.3, 22. , 20.1,
 22.2, 23.7, 17.6, 18.5, 24.3, 20.5, 24.5, 26.2, 24.4, 24.8, 29.6,
 42.8, 21.9, 20.9, 44. , 50. , 36. , 30.1, 33.8, 43.1, 48.8, 31. ,
 36.5, 22.8, 30.7, 50. , 43.5, 20.7, 21.1, 25.2, 24.4, 35.2, 32.4,
 32. , 33.2, 33.1, 29.1, 35.1, 45.4, 35.4, 46. , 50. , 32.2, 22. ,
 20.1, 23.2, 22.3, 24.8, 28.5, 37.3, 27.9, 23.9, 21.7, 28.6, 27.1,
 20.3, 22.5, 29. , 24.8, 22. , 26.4, 33.1, 36.1, 28.4, 33.4, 28.2,
 22.8, 20.3, 16.1, 22.1, 19.4, 21.6, 23.8, 16.2, 17.8, 19.8, 23.1,
 21. , 23.8, 23.1, 20.4, 18.5, 25. , 24.6, 23. , 22.2, 19.3, 22.6,
 19.8, 17.1, 19.4, 22.2, 20.7, 21.1, 19.5, 18.5, 20.6, 19. , 18.7,
 32.7, 16.5, 23.9, 31.2, 17.5, 17.2, 23.1, 24.5, 26.6, 22.9, 24.1,
 18.6, 30.1, 18.2, 20.6, 17.8, 21.7, 22.7, 22.6, 25. , 19.9, 20.8,
 16.8, 21.9, 27.5, 21.9, 23.1, 50. , 50. , 50. , 50. , 50. , 13.8,
 13.8, 15. , 13.9, 13.3, 13.1, 10.2, 10.4, 10.9, 11.3, 12.3, 8.8,
 7.2, 10.5, 7.4, 10.2, 11.5, 15.1, 23.2, 9.7, 13.8, 12.7, 13.1,
 12.5, 8.5, 5. , 6.3, 5.6, 7.2, 12.1, 8.3, 8.5, 5. , 11.9,
 27.9, 17.2, 27.5, 15. , 17.2, 17.9, 16.3, 7. , 7.2, 7.5, 10.4,
 8.8, 8.4, 16.7, 14.2, 20.8, 13.4, 11.7, 8.3, 10.2, 10.9, 11. ,
 9.5, 14.5, 14.1, 16.1, 14.3, 11.7, 13.4, 9.6, 8.7, 8.4, 12.8,
 10.5, 17.1, 18.4, 15.4, 10.8, 11.8, 14.9, 12.6, 14.1, 13. , 13.4,
 15.2, 16.1, 17.8, 14.9, 14.1, 12.7, 13.5, 14.9, 20. , 16.4, 17.7,
 19.5, 20.2, 21.4, 19.9, 19. , 19.1, 19.1, 20.1, 19.9, 19.6, 23.2,
 29.8, 13.8, 13.3, 16.7, 12. , 14.6, 21.4, 23. , 23.7, 25. , 21.8,
 20.6, 21.2, 19.1, 20.6, 15.2, 7. , 8.1, 13.6, 20.1, 21.8, 24.5,
 23.1, 19.7, 18.3, 21.2, 17.5, 16.8, 22.4, 20.6, 23.9, 22. , 11.9])
```

```
In [18]: x=boston_data.data
y=boston_data.target
knn= KNeighborsRegressor(n_neighbors=10)
y_pred = cross_val_predict(knn, x, y, cv=5)
```

```
In [19]: y_pred
```

```
Out[19]: array([24.   , 26.17, 21.95, 23.09, 21.99, 21.95, 23.72, 22.32, 23.54,
 27.12, 24.93, 22.62, 23.56, 22.63, 31.44, 23.75, 27.76, 27.38,
 25.43, 24.88, 27.33, 22.32, 22.32, 24.54, 22.32, 29.31, 31.44,
 29.31, 26.45, 31.44, 31.52, 28.19, 20.09, 34.18, 20.09, 21.17,
 23.19, 23.67, 24.47, 35.55, 33.77, 23.99, 23.99, 23.99, 21.66,
 21.65, 21.65, 30.28, 25.19, 21.95, 30.77, 31.98, 30.06, 30.06,
 31.11, 39.74, 25.23, 37.2 , 25.35, 25.61, 28.85, 36.17, 29.06,
 25.96, 28.66, 26.8 , 26.38, 27.11, 26.43, 26.43, 28.47, 28.32,
 30.27, 28.09, 25.67, 20.11, 28.54, 20.11, 20.11, 20.36, 26.85,
 32.41, 26.89, 26.01, 22.82, 25.55, 22.88, 22.79, 21.5 , 21.88,
 21.35, 21.68, 31.67, 31.17, 37.68, 25.85, 21.91, 21.81, 24.1 ,
 22.14, 19.22, 19.68, 13.62, 21.05, 21.05, 21.05, 21.05, 21.05,
 21.05, 21.05, 21.29, 20.52, 20.52, 20.52, 20.52, 19.2 , 19.34,
 20.52, 19.74, 19.6 , 28.03, 28.82, 23.98, 23.98, 25.38, 25.66,
 26.64, 20.52, 20.52, 20.52, 20.52, 20.52, 20.52, 20.52, 17.69,
 20.52, 20.52, 20.52, 20.52, 20.52, 20.52, 20.52, 21.05, 21.05,
 21.05, 15.84, 15.84, 21.05, 20.52, 20.52, 20.52, 20.52, 21.01,
 17.52, 20.61, 14.56, 14.12, 20.52, 20.52, 20.52, 20.54, 20.52,
 20.52, 20.52, 21.05, 17.52, 20.52, 17.52, 19.01, 20.54, 18.68,
 20.52, 19.76, 22.62, 21.41, 24.47, 22.63, 23.14, 27.03, 27.68,
 27.56, 27.68, 26.56, 26.56, 25.66, 30.5 , 29.84, 23.14, 23.04,
 23.27, 23.82, 23.27, 23.2 , 30.17, 30.33, 32.71, 21.91, 22.38,
 21.91, 24.01, 24.01, 22.99, 22.39, 33.36, 33.36, 26.65, 23.89,
 21.7 , 23.26, 20.11, 22.79, 22.57, 22.76, 25.79, 22.84, 25.02,
 23.55, 22.5 , 22.79, 22.79, 19.17, 18.24, 20.79, 20.46, 20.95,
 20.94, 20.59, 19.37, 23.86, 23.71, 21.1 , 19.73, 21.24, 21.7 ,
 18.18, 21.29, 20.56, 21.97, 22.67, 23.45, 23.36, 23.08, 22.33,
 22.93, 19.43, 20.16, 21.21, 19.43, 20.9 , 21.87, 21.87, 22.15,
 21.92, 21.92, 24.6 , 26.42, 33.36, 22.32, 22.32, 22.32, 22.81,
 22.32, 22.32, 22.32, 22.32, 23.53, 20.46, 22.35, 23.85, 26.5 ,
 24.04, 24.21, 28.17, 27.15, 26.51, 24.49, 25.28, 25.23, 26.51,
 23.19, 28.17, 25.4 , 27.72, 33.75, 32.46, 25.3 , 33.36, 24.29,
 23.66, 25.5 , 32.85, 32.85, 32.85, 26.23, 23.58, 25.73, 22.17,
 21.68, 26.73, 26.6 , 27.36, 21.21, 21.28, 21.39, 31.6 , 30.14,
 29.47, 29.47, 28.95, 25.06, 27.43, 22.78, 20.85, 28.24, 22.74,
 24.76, 21.04, 24.74, 26.14, 23.07, 24.16, 23.55, 24.16, 24.52,
 26.89, 25.75, 26.88, 23.84, 21.77, 24.15, 21.63, 23.43, 24.7 ,
 28.56, 28.56, 27.29, 25.84, 28.69, 27.29, 25.84, 28.69, 22.95,
 20.74, 29.09, 30.1 , 22.21, 21.14, 27.54, 30.96, 24.12, 23.77,
 31.57, 31.99, 40.24, 26.72, 27.77, 16.3 , 18.07, 20.79, 20.79,
 18.27, 16.56, 17.25, 18.12, 19.07, 18.12, 17.09, 14.   , 17.25,
 17.25, 16.79, 15.53, 16.77, 14.83, 14.83, 15.74, 15.9 , 15.49,
 15.9 , 14.83, 13.54, 15.74, 15.33, 16.05, 15.22, 14.83, 15.36,
 16.05, 15.26, 16.46, 16.46, 18.94, 15.33, 17.2 , 15.74, 16.3 ,
 16.46, 16.05, 13.54, 17.01, 15.36, 15.74, 16.3 , 15.9 , 16.21,
 9.93, 22.13, 23.67, 19.55, 20.47, 14.53, 14.53, 14.53, 20.47,
 18.4 , 14.53, 14.53, 20.47, 14.53, 14.53, 19.55, 19.55, 19.55,
 14.53, 14.53, 14.53, 14.53, 14.53, 20.47, 14.53, 18.4 , 18.4 ,
 20.47, 20.47, 20.47, 20.47, 14.53, 14.53, 18.28, 12.07, 10.09,
 13.58, 12.41, 12.01, 20.47, 14.53, 19.55, 12.94, 11.75, 19.55,
 14.53, 24.01, 16.91, 26.29, 14.53, 14.53, 14.53, 14.53, 19.55,
 15.42, 19.55, 16.35, 15.42, 19.27, 17.67, 20.99, 14.53, 23.67,
 23.86, 17.67, 15.42, 15.42, 18.24, 25.22, 24.01, 19.55, 13.35,
```

```
22.86, 16.75, 19.34, 21.93, 21.93, 21.93, 19.01, 20.76, 19.01,  
16.59, 19.01, 14.45, 23.84, 19.55, 13.31, 15.42, 21.63, 22. ,  
20.93, 21.51, 21.56, 21.89, 21.56, 21.47, 24.98, 24.56, 24.1 ,  
24.1 , 24.2 ])
```

```
In [20]: x=boston_data.data  
y=boston_data.target  
knn=KNeighborsRegressor(n_neighbors=10)  
y_pred =cross_val_predict(knn, x, y, cv=5)
```

```
In [21]: y_pred
```

```
Out[21]: array([24.   , 26.17, 21.95, 23.09, 21.99, 21.95, 23.72, 22.32, 23.54,
 27.12, 24.93, 22.62, 23.56, 22.63, 31.44, 23.75, 27.76, 27.38,
 25.43, 24.88, 27.33, 22.32, 22.32, 24.54, 22.32, 29.31, 31.44,
 29.31, 26.45, 31.44, 31.52, 28.19, 20.09, 34.18, 20.09, 21.17,
 23.19, 23.67, 24.47, 35.55, 33.77, 23.99, 23.99, 23.99, 21.66,
 21.65, 21.65, 30.28, 25.19, 21.95, 30.77, 31.98, 30.06, 30.06,
 31.11, 39.74, 25.23, 37.2 , 25.35, 25.61, 28.85, 36.17, 29.06,
 25.96, 28.66, 26.8 , 26.38, 27.11, 26.43, 26.43, 28.47, 28.32,
 30.27, 28.09, 25.67, 20.11, 28.54, 20.11, 20.11, 20.36, 26.85,
 32.41, 26.89, 26.01, 22.82, 25.55, 22.88, 22.79, 21.5 , 21.88,
 21.35, 21.68, 31.67, 31.17, 37.68, 25.85, 21.91, 21.81, 24.1 ,
 22.14, 19.22, 19.68, 13.62, 21.05, 21.05, 21.05, 21.05, 21.05,
 21.05, 21.05, 21.29, 20.52, 20.52, 20.52, 20.52, 19.2 , 19.34,
 20.52, 19.74, 19.6 , 28.03, 28.82, 23.98, 23.98, 25.38, 25.66,
 26.64, 20.52, 20.52, 20.52, 20.52, 20.52, 20.52, 20.52, 17.69,
 20.52, 20.52, 20.52, 20.52, 20.52, 20.52, 20.52, 21.05, 21.05,
 21.05, 15.84, 15.84, 21.05, 20.52, 20.52, 20.52, 20.52, 21.01,
 17.52, 20.61, 14.56, 14.12, 20.52, 20.52, 20.52, 20.54, 20.52,
 20.52, 20.52, 21.05, 17.52, 20.52, 17.52, 19.01, 20.54, 18.68,
 20.52, 19.76, 22.62, 21.41, 24.47, 22.63, 23.14, 27.03, 27.68,
 27.56, 27.68, 26.56, 26.56, 25.66, 30.5 , 29.84, 23.14, 23.04,
 23.27, 23.82, 23.27, 23.2 , 30.17, 30.33, 32.71, 21.91, 22.38,
 21.91, 24.01, 24.01, 22.99, 22.39, 33.36, 33.36, 26.65, 23.89,
 21.7 , 23.26, 20.11, 22.79, 22.57, 22.76, 25.79, 22.84, 25.02,
 23.55, 22.5 , 22.79, 22.79, 19.17, 18.24, 20.79, 20.46, 20.95,
 20.94, 20.59, 19.37, 23.86, 23.71, 21.1 , 19.73, 21.24, 21.7 ,
 18.18, 21.29, 20.56, 21.97, 22.67, 23.45, 23.36, 23.08, 22.33,
 22.93, 19.43, 20.16, 21.21, 19.43, 20.9 , 21.87, 21.87, 22.15,
 21.92, 21.92, 24.6 , 26.42, 33.36, 22.32, 22.32, 22.32, 22.81,
 22.32, 22.32, 22.32, 22.32, 23.53, 20.46, 22.35, 23.85, 26.5 ,
 24.04, 24.21, 28.17, 27.15, 26.51, 24.49, 25.28, 25.23, 26.51,
 23.19, 28.17, 25.4 , 27.72, 33.75, 32.46, 25.3 , 33.36, 24.29,
 23.66, 25.5 , 32.85, 32.85, 32.85, 26.23, 23.58, 25.73, 22.17,
 21.68, 26.73, 26.6 , 27.36, 21.21, 21.28, 21.39, 31.6 , 30.14,
 29.47, 29.47, 28.95, 25.06, 27.43, 22.78, 20.85, 28.24, 22.74,
 24.76, 21.04, 24.74, 26.14, 23.07, 24.16, 23.55, 24.16, 24.52,
 26.89, 25.75, 26.88, 23.84, 21.77, 24.15, 21.63, 23.43, 24.7 ,
 28.56, 28.56, 27.29, 25.84, 28.69, 27.29, 25.84, 28.69, 22.95,
 20.74, 29.09, 30.1 , 22.21, 21.14, 27.54, 30.96, 24.12, 23.77,
 31.57, 31.99, 40.24, 26.72, 27.77, 16.3 , 18.07, 20.79, 20.79,
 18.27, 16.56, 17.25, 18.12, 19.07, 18.12, 17.09, 14.   , 17.25,
 17.25, 16.79, 15.53, 16.77, 14.83, 14.83, 15.74, 15.9 , 15.49,
 15.9 , 14.83, 13.54, 15.74, 15.33, 16.05, 15.22, 14.83, 15.36,
 16.05, 15.26, 16.46, 16.46, 18.94, 15.33, 17.2 , 15.74, 16.3 ,
 16.46, 16.05, 13.54, 17.01, 15.36, 15.74, 16.3 , 15.9 , 16.21,
 9.93, 22.13, 23.67, 19.55, 20.47, 14.53, 14.53, 14.53, 20.47,
 18.4 , 14.53, 14.53, 20.47, 14.53, 14.53, 19.55, 19.55, 19.55,
 14.53, 14.53, 14.53, 14.53, 14.53, 20.47, 14.53, 18.4 , 18.4 ,
 20.47, 20.47, 20.47, 20.47, 14.53, 14.53, 18.28, 12.07, 10.09,
 13.58, 12.41, 12.01, 20.47, 14.53, 19.55, 12.94, 11.75, 19.55,
 14.53, 24.01, 16.91, 26.29, 14.53, 14.53, 14.53, 14.53, 19.55,
 15.42, 19.55, 16.35, 15.42, 19.27, 17.67, 20.99, 14.53, 23.67,
 23.86, 17.67, 15.42, 15.42, 18.24, 25.22, 24.01, 19.55, 13.35,
 22.86, 16.75, 19.34, 21.93, 21.93, 21.93, 19.01, 20.76, 19.01,
```

```
16.59, 19.01, 14.45, 23.84, 19.55, 13.31, 15.42, 21.63, 22. ,
20.93, 21.51, 21.56, 21.89, 21.56, 21.47, 24.98, 24.56, 24.1 ,
24.1 , 24.2 ])
```

```
In [22]: from math import sqrt
```

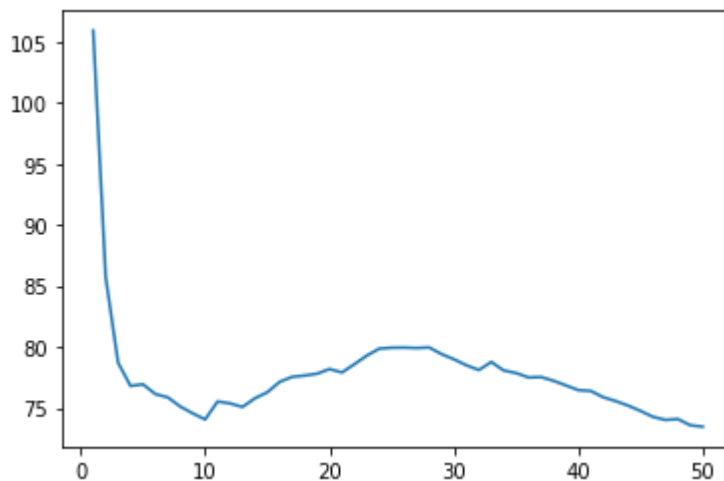
```
In [23]: print(mean_squared_error(y,y_pred))
print(r2_score(y,y_pred))
```

```
74.05401304347826
0.12278604134701132
```

```
In [24]: error = []
for k in range(1,51):
    knn=KNeighborsRegressor(n_neighbors=k)
    y_pred = cross_val_predict(knn, x, y, cv=5)
    error.append(mean_squared_error(y,y_pred))
```

```
In [25]: plt.plot(range(1,51),error)
```

```
Out[25]: [<matplotlib.lines.Line2D at 0x1eaffd81f40>]
```



```
In [26]: from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make_pipeline
```

```
In [27]: classifier_pipeline = make_pipeline(StandardScaler(), KNeighborsRegressor(n_neigh
```

```
In [28]: y_pred=cross_val_predict(classifier_pipeline, x, y, cv=5)
print(mean_squared_error(y,y_pred))
print(r2_score(y,y_pred))
```

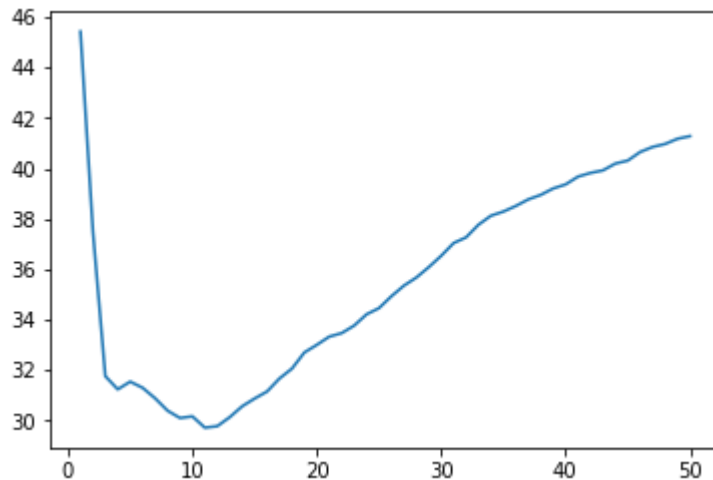
```
30.168073715415016
0.6426411712043607
```



```
In [29]: error = []
         for k in range(1,51):
             classifier_pipeline= make_pipeline(StandardScaler(), KNeighborsRegressor(n_ne
             y_pred = cross_val_predict(classifier_pipeline, x, y, cv=5)
             error.append(mean_squared_error(y,y_pred))
```

```
In [30]: plt.plot(range(1,51), error)
```

```
Out[30]: [<matplotlib.lines.Line2D at 0x1eaffe68580>]
```



```
In [31]: from sklearn.model_selection import KFold
```

```
In [32]: cv =KFold(n_splits=5 ,random_state=0, shuffle=False)
```

C:\Users\Daren Pava\anaconda3\lib\site-packages\sklearn\model_selection_split.py:293: FutureWarning: Setting a random_state has no effect since shuffle is False. This will raise an error in 0.24. You should leave random_state to its default (None), or set shuffle=True.
 warnings.warn(

```
In [33]: classifier_pipeline=make_pipeline(StandardScaler(),KNeighborsRegressor(n_neighbor
y_pred = cross_val_predict(classifier_pipeline, x, y, cv=cv)
print(mean_squared_error(y,y_pred))
print(r2_score(y,y_pred))
```

```
30.168073715415016
0.6426411712043607
```

```
In [34]: ## Presentado por Daren Pava
```

```
In [ ]:
```

